Strategy Building Software for Atmospheric Retrievals

Sassaneh Hojaiji Poosti
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr., Pasadena, CA 91109
(818)354-3770
poosti@tes-mail.jpl.nasa.gov

TABLE OF CONTENTS

Acknowledgements

I would like to extend my appreciation to my colleges at Jet Propulsion Laboratory for supporting me in producing this work. I would also like to thank Kenneth Scott for his help in preparation of some of the data flow diagrams.

This work was performed at the Jet propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Abstract - Atmospheric retrieval consists of a series of the scientific algorithms performed to retrieve the actual state of the atmosphere in terms of its temperature and chemical constituents. These scientific algorithms include calculations that convert the radiances observed by the instruments to geophysical quantities and profiles of atmospheric constituents.

Since most atmospheric retrieval algorithms are computationally expensive, and the success of the retrieval depends very much on the input, it is important to provide the best input that will result in the best atmospheric profiles. The Strategy Building software selects the best input for the retrieval of various target scenes based on specific strategy selection rules that are provided by atmosphere scientists. It improves the robustness and flexibility of the atmospheric retrieval software and allows the researchers implement various ideas even after instrument's launch with the minimal cost. The strategy selection logics also incorporate the issues of the retrieval of the atmosphere of one target scene in a sequence of retrievals. Static and dynamic strategy selections will be discussed in this paper.

Object-oriented development techniques are used in this paper. Using this approach adds a great deal of modifiability which is a necessity and the objective of Strategy Building. The presented design in this paper allows it to be reused in development of many atmospheric Retrieval software, to improve their maintainability.

Using the Strategy Building software adds to the lifetime of the software and provides flexibility. This makes the software more profitable in the way that it can be used for a longer time, for more research and the produced output is more reliable and it also mitigates the needs for development of extra software for trying different ideas in research.

1. INTRODUCTION

In the area of atmospheric science research, majority of the software have been developed with conventional approaches. Many of them are hard to maintain and to be reused. There are new software development procedures that produce more maintainable and more reusable research software. One of the approaches that makes this possible is object-oriented software development.

Modifiability is another feature of the good software development that many of the atmospheric research software can benefit from. Such software are being developed to do research around new ideas. In these cases, since new theories, algorithms, and/or input are being used, there are many unknowns involved. Therefore, the software is developed with the minimum known requirements, and for this reason many changes may be anticipated after the software is used for a while and the produced output is analyzed. Therefore, modifiability is the important feature that can save the project a lot of money while allowing the researchers try various methods and ideas using the same software as opposed to the limited flexibility that they'll face otherwise.

Strategy Builder software is able to provide the mentioned features, modifiability and reusability. The objective of this paper is to introduce the effort that has been made for Tropospheric Emission Spectrometer (TES) at JPL to develop this software and discuss the potentials of its reuse in other atmospheric retrieval software developments. TES is one of the first NASA projects developing its science data processing software using object-oriented technology. This provides the opportunity to develop software that can be reused by the projects with similar algorithms and objectives. This paper is based on the work done for the Graduate project at California State University, Northridge in association with the TES data processing software system.

Tropospheric Emission Spectrometer

TES is one of the instruments of the EOS Aura mission that is a part of the NASA's Earth Science Enterprise program. This program consists of missions that will help scientists to observe and understand the changes of natural processes and their effects on Earth. Oceans, clouds, land[1] surface, and chemistry of the atmosphere are the major areas of study in these projects. TES will provide the global concentration map of the atmospheric chemicals in troposphere layer and its focus will be on the

0-7803-6599-2/01/$10.00 © 2001 IEEE

concentration map of the atmospheric chemicals in troposphere layer and its focus will be on the global distribution of tropospheric ozone and the factors that affect ozone formation.

Atmospheric Retrieval

Atmospheric retrieval refers to the scientific algorithms performed to retrieve the state of the atmosphere in terms of its temperature, constituents, etc. These algorithms include parts for converting the radiances from the atmosphere gases, received by the instruments, to quantities such as parts per volume of the gas. The resulted quantities are then available to other scientists for studies of the various characteristics of the Earth system.

TES Data Processing

TES uses the following iterative process for retrievals:

1- We guess a profile.
2- We determine the weighting functions.
3- The "Forward Model" algorithm provides the radiance with respect to frequency.
4- If the radiance computed from the guess profile matches the real scanned radiance, then we decide to keep the guess profile as the real profile.
5- If not, the guess profile is adjusted until the deference between the real radiance data and the computed data is small.
6- We repeat steps 3 to 5 until we find a match to the radiance data.

TES data processing consists of the following levels: At Level 1A the raw data from the spacecraft are decommutated and the interferograms reconstructed. At Level 1B, the interferograms are converted to calibrated spectra. At Level 2, vertical concentration profiles of the selected species are extracted from the data through the process of retrieval [6].

2. TES RETRIEVAL ALGORITHM

The complete retrieval of the state of the atmosphere at each target scene requires sequential performances of the above retrieval where at each retrieval, a specific set of the atmospheric elements will be retrieved. Figure 1 shows the loop that carries out the sequential retrievals in TES Level 2 software. Each Retrieval in this loop is called a Retrieval Step.

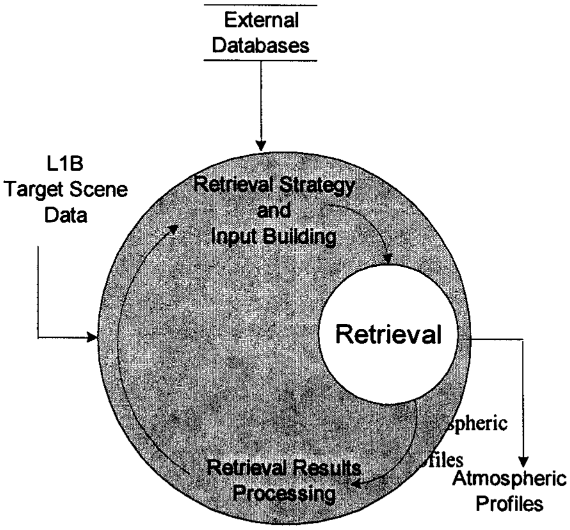


Figure 1 – Loop of sequential retrievals

In this loop in a sequence of Strategy and input building, Retrieval, and Results processing take place. The series of Retrieval Steps that produce the complete profiles of the atmospheric constituents is called Retrieval Sequence. The number of times that this loop will be executed is the number of Retrieval Steps in the Retrieval Sequence. After each Retrieval Step the results will be analyzed, and based on those results the strategy for the next step will be built. I would like to emphasize on the concept that the strategy of each Retrieval Step may be built dynamically and based on any relevant parameter including the results of another previously done retrieval.

3. STRATEGY BUILDER

Improving the robustness

Flexibility and robustness are particularly beneficial in atmospheric research software such as TES Level 2 software. The algorithm as described earlier starts with an initial guess for the value of the retrieval element. The initial guess value is the closest value to the actual value of the element among all available data for that element. But it is known that this closest value is not as accurate as desired. As the whole purpose of the scientific project is to obtain the accurate values for these elements, Retrieval software's output is supposed to be an accurate value. Since the algorithms used in the Retrieval software are being used for the project's specific purpose for the first time, it is not proven that the algorithms will result in the desired accurate values. The only time that this may be proven is after the spacecraft is launched. At this time the developed software will be used to process the real atmospheric data received from the instrument instead of simulated data. Therefore, the modifications to the algorithm software might seem necessary in order to obtain more accurate data. Needless to say, the modifications can be so extensive that the project will not be able to completely implement them. In such a situation, the scientist will have to use less accurate data for their research.

Parameterization helps in achieving more flexibility in the software. It is possible to have a dynamic and intelligent system to provide appropriate values for parameters in various situations. This can be accomplished by developing Strategy Builder software. It is an additional component to the architecture for packaging the parameters and their providers by which greater encapsulation and higher modifiability of the software is made possible.

Strategy

There is a set of parameters that determine the way that the atmosphere of a particular target scene should be retrieved. Strategy refers to the entity that contains the values for the parameters. The collection of these values for a specific set of Retrieval parameters in called the "Strategy" for a specific retrieval. The specified strategy for a particular Retrieval, then will be used to build the input for that Retrieval.

The objective of the program that builds the strategy (Strategy Builder program), is 1) to select the best suitable source for obtaining the parameters' values and 2) return these data values from the selected sources. Then these values will be put together to form the Strategy. The Strategy for one retrieval program can be thought of as a set of "name=value" pairs.

The complete retrieval of the atmosphere is consisted of a set of consecutive retrievals, and each retrieval in this sequence is called a Retrieval Step. The strategy for the entire retrieval sequence may be presented as a table with the retrieval parameter as its fields and the retrieval steps as its rows. Strategy in the form of tables is discussed in detail and examples of Strategy are given later in this paper.

Parameterization of Retrieval

Some of the identified retrieval parameters in TES are Shown in Table 1.

Table 1 – Examples of retrieval parameters in TES retrieval algorithm

The retrieval element	Obtaining the atmospheric profiles is better done when they are being retrieved by a sequence of retrievals as opposed to retrieving them all at the same time. The combination of the species for each retrieval is one of the parameters.
The retrievals order	The results of one retrieval will be used in the next retrieval. Therefore, the order in which the sequential retrievals are taking place is another parameter.
The retrieval type	The retrieval type is another parameter that can be different in each retrieval. The types are normal, shape retrieval, and the type in which the calculations of jacobians, radiances, and transmittances in previous retrievals are reused in the current retrieval.
Mapping and inverse mapping specifications	Another element that the software allows to be different for each retrieval is the functions that map the retrieval levels to the reporting grid levels and vice versa.
The initial guess	The initial guess value for the Volume Mixing Ratio of every retrieval element such as Temperature or Water Vapor can be derived from different external climatology and meteorology sources. It is also possible to use previously processed data as an initial guess for a new retrieval process. This is in fact the most desirable initial guess whenever possible and available.
The a priori data	The a priori data usually comprise of climatological estimate of the state of the atmosphere plus some amount of uncertainty. It is usually expressed as a covariance matrix.

Some of these parameters such as the a priori and initial guess data exist in the algorithms used by other NASA instruments such as MIPAS [7], HIRDLS [8].

Dynamic Strategy Selection

Providing dynamic values for each parameter is one of the goals of the Strategy Builder. We want to be able to choose the best possible value for the parameter based on the current state. For instance, we obtain the initial guess value for H_2O from different sources based on some particular set of logic. Since every parameter can have such logic, simple or complicated, a provider of the parameter value will carry out the task of providing the best value for the parameter based on that logic. As an example of the logic for selecting one of these parameters, the flowchart for initial guess selection logic is shown in figure 2.

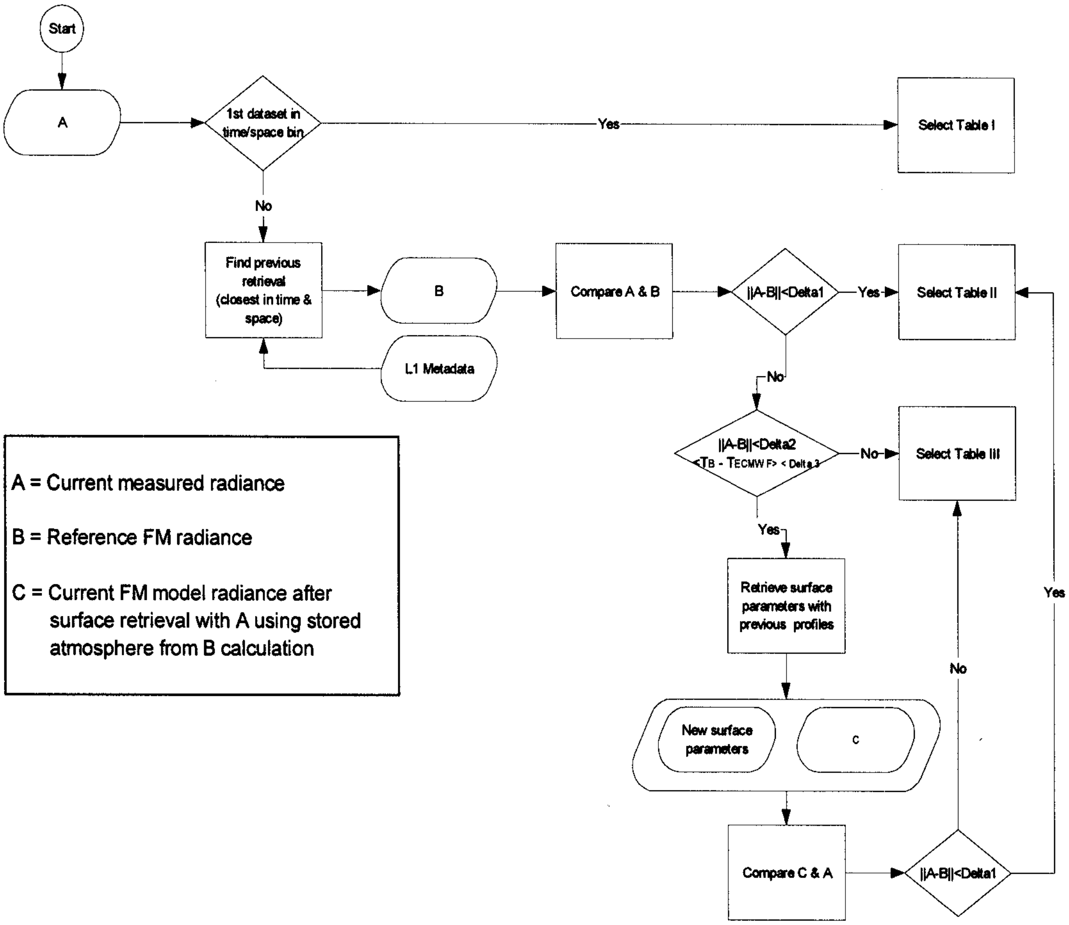

Start
A
1st dataset in time/space bin
Yes
Select Table I
No
Find previous retrieval (closest in time & space)
B
Compare A & B
||A-B||<Delta1
Yes
Select Table II
L1 Metadata
No
||A-B||<Delta2
<TB - TECMW F> < Delta3
No
Select Table III
Yes
Retrieve surface parameters with previous profiles
New surface parameters
C
Compare C & A
||A-B||<Delta1
No
Yes
A = Current measured radiance
B = Reference FM radiance
C = Current FM model radiance after surface retrieval with A using stored atmosphere from B calculation

Figure 2 - Flowchart for initial guess selection logic

Strategy Selection

In the systems like this, we want to give the chance of alternative strategies to the retrievals. In case a selected strategy did not work, the system should try another strategy based on what it has learned from the first one and probably some predefined selection rules.

Intelligence in Strategy Selection

It is desirable to add intelligence to the Strategy Selection schemes constantly. The system can learn more about strategies as it continues to process the Target Scenes. As mentioned earlier, the produced output of the retrieval algorithm is the set of profiles of all retrieved chemicals in the atmosphere such as O_3, H_2O, etc. at each Target Scene. Each profile is an array of the amounts of one species at different levels of one column of atmosphere at that Target Scene. After the retrieval of a particular Target Scene is done, the output can be compared to the results of the other retrievals and analyzed. To do this, the analysis code will use a database of previous results and the strategies used in getting these output. Performing this scientific analysis will help improving the strategy selection schemes, so that better values be selected for the program's parameters in future retrievals.

Sequential Retrieval of the Atmosphere

One of the results of the analysis of Strategy Builder problem was that the retrieval of the atmospheric profiles of each Target Scene would not be done in one step. We will have a sequence of retrievals each performed on a specific set of atmospheric molecules. This sequence is called Retrieval Sequence, and each retrieval of this sequence is called Retrieval Step. Each retrieval will be done separately, although the sequence follows a particular order and each Retrieval Step depends on the results of the previous one.

Domain Objects

Reusability of the developed Strategy Builder has been one of objectives of this research. By domain analysis [5] & [6], common reusable classes and objects can be identified for the projects within this particular domain. The following domain objects are identified as the result of the investigating the problem and the problem domain:

1. Retrieval Algorithm
This object can carry out retrieving a set of elements simultaneously

2. Retrieval Sequence
This object encapsulates the information about a sequence of retrievals that altogether complete the retrieval of constituents of one column of atmosphere at a particular geolocation.

3. Retrieval Step
When the retrievals are divided into several steps, this object can encapsulate the information about each step.

4. Sequence Strategy
The list of parameters values that need to be known in advance, in order to be able to specify a sequence of retrieval steps and the Strategy parameters' values for each step.

5. Step Strategy
The strategy parameters' values for one retrieval step.

6. Sequence Controller
This object is responsible for controlling the loop of the retrievals. It contains the attributes that are important in this control and the operations that perform the control.

7. Reusable Calculations
Reusable Calculations can be an aggregate class with the reusable aggregations that will be a part of the Retrieval Input.

8. Level 1 Data
The Level 1 spectrum data that is input to the Strategy Builder.

9. Strategy Parameter
The data structure that contains the attributes of a Retrieval algorithm parameter. The object can also include the supplier method for parameter values.

10. Parameter Supplying Algorithm
In case of multiple implementations of the supplier algorithm, it can be generalized in an abstract class and implemented in inherited objects for every specific usage.

11. Source Data Store
The information about a data source that is encapsulated to be used by the supplier algorithm.

12. Initial Guess Data
The data structure containing the initial profiles of the retrieval elements.

These objects are several candidates for reuse in the atmospheric retrieval software. However, the domain analysis of the atmospheric retrievals can certainly result many other domain objects, which is a good research effort in the context of reuse in atmospheric science software [5]. The list presented in this paper contains only the objects related to the Strategy Builder software for retrieval algorithms.

Strategy Builder's Architecture

Strategy Builder's high level architecture is shown in figure 3. Databases are mostly used for the data store components such as L1B data and atmospheric profiles in this architecture. The Retrieval Analyzer is the component that is able to analyze the recorded results from the previous retrievals and update the strategies. [3], [4].

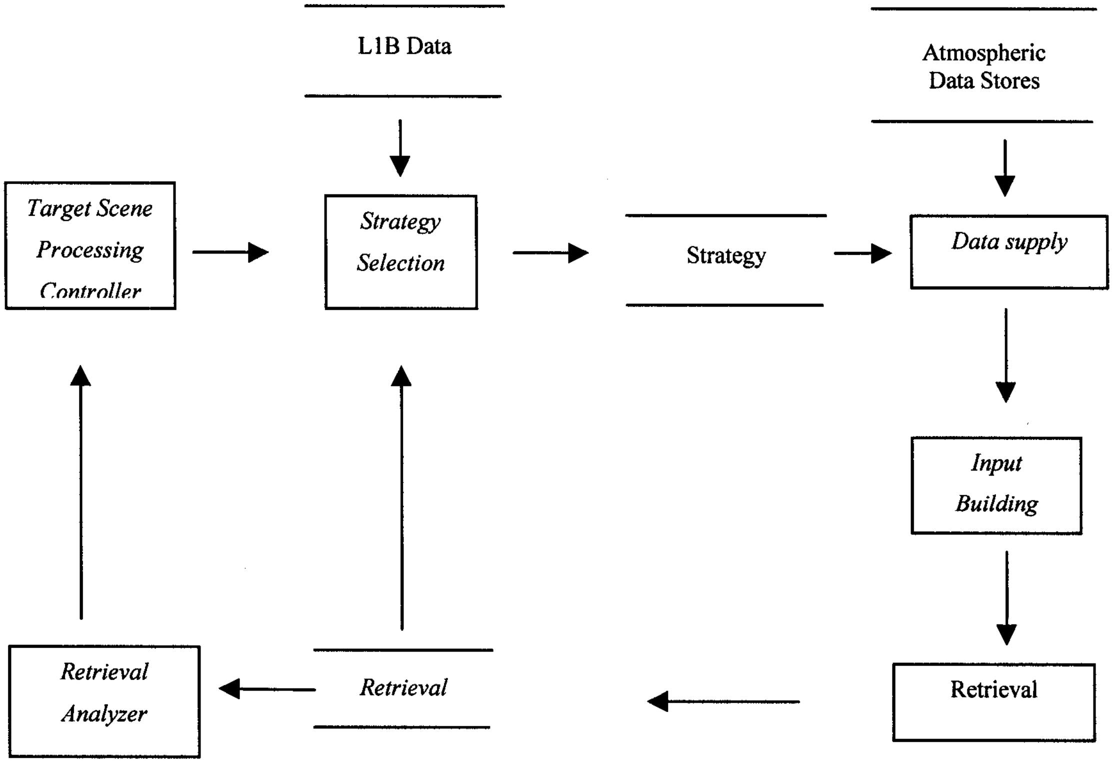
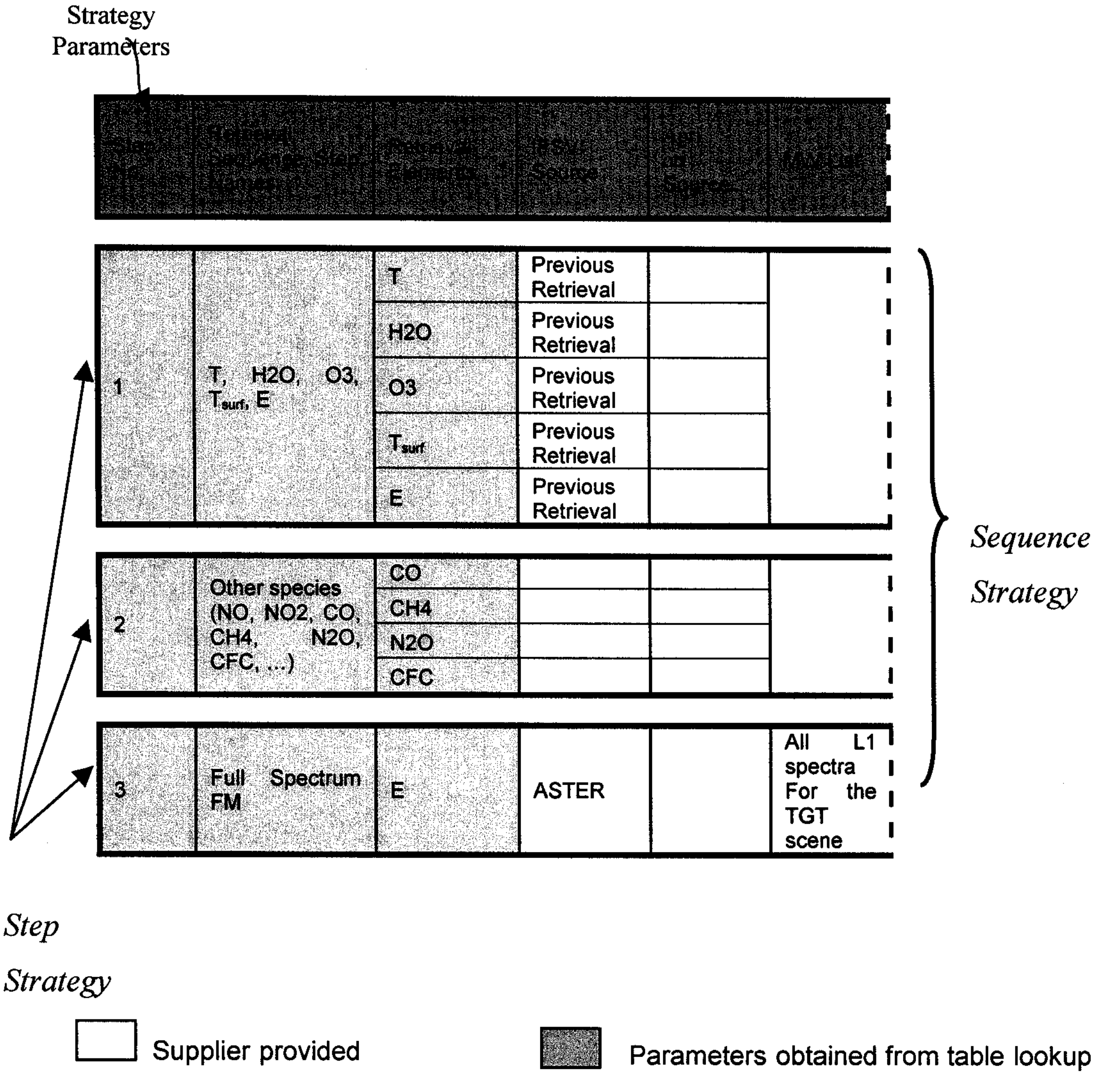


Figure 3 – Strategy Builder's Architecture

4. STRATEGY BUILDER CONCEPTS

Based on the results of the analysis effort, tables seemed to be the most appropriate structure for demonstrating a typical strategy. Tables are able to communicate well the concepts such as the Sequence Strategy, Step Strategy, Strategy Parameters, etc. as shown in the picture below.



Step No.	Retrieval Sequence Step Names	Retrieval Elements	[illegible] Source	[illegible] Source	[illegible]
1	T, H2O, O3, T_{surf}, E	T	Previous Retrieval		
		H2O	Previous Retrieval		
		O3	Previous Retrieval		
		T_{surf}	Previous Retrieval		
		E	Previous Retrieval		
2	Other species (NO, NO2, CO, CH4, N2O, CFC, ...)	CO			
		CH4			
		N2O			
		CFC			
3	Full Spectrum FM	E	ASTER		All L1 spectra For the TGT scene

□ Supplier provided ▨ Parameters obtained from table lookup

Figure 4 – Strategy table components

By means of the tables the scientists were able to have a clear picture of the potential variables, which would construct the parameter list of the Retrieval Strategies in the software. Tables also gave a better opportunity to study the parameters further. They were also used for analyzing the dependencies between the parameters, as well as the relationships between the parameter's selection rules and the Retrieval algorithm.

The table's fields are Strategy parameters of the Retrieval algorithm. Each row in the table contains the strategy parameters values for one specific Retrieval Step. The number of rows can be different for each Target Scene. The table as a whole is the Target Scene's Retrieval Strategy. It is a structure that contains the parameter values of the entire retrieval of a Target Scene. One example of the strategy tables is shown in figure 5.

Step No.	Retrieval Sequence Step Names	Retrieval Elements	IFSV Source	Retrieval Levels	A priori Source	Retrieval Type	Max No. of Iterations	Inversion Method	Continuation Conditions
1	T, T_{surf}, E	T	ECMWF	L1, ...	ECMWF	Normal	3	MAP/GN	Convergence
		T_{surf}	ECMWF		ECMWF				
		E	ASTER						
2	H_2O	H_2O	ECMWF	L1, ...	ECMWF	Normal	3	MAP/GN	Convergence
3	O_3 initial guess refinement	O_3	OMI, Climatology	Profile Shape & Offset params.	Climatology	Shape retrieval	2	ML/GN	-
4	O_3	O_3	Previous Step	L1, ...	Climatology	Normal	2	MAP/GN	-
5	T, H_2O, O_3, T_{surf}, , E	T	Previous Step	L1, ...		Normal	2	MAP/GN	-
		H_2O	Previous Step	L1, ...					
		O_3	Previous Step	L1, ...					
		T_{surf}	Previous Step						
		E	ASTER						
6	Other species (NO, NO_2, CO, CH_4, N_2O, CFC, ...)	CO		L1, ...		Normal	2	MAP/GN	-
		CH_4		L1, ...					
		N_2O		L1, ...					
		CFC		L1, ...					
7	Full Spectrum FM	E	ASTER			Surface only	1	MAP/GN	Stop
		T	Previous Step						

Figure 5 - Retrieval Strategy Table for Nadir retrieval parameters
Case of 1st data set in time/space bin

Static Strategy Tables versus Dynamic Strategy Selection

In TES there are different strategies for nadir and limb cases. Therefore, the view mode can be one of the strategy selection parameters. The other strategy selection parameter relates to the reusability of the radiance calculations. In some cases, the strategy specifies the use of some previous calculations. In this case the reusability flag will be set.

The static Strategy Tables fit to the requirements of TES software very well. In this case, the scientists provide several Strategy Tables, each suitable for some particular cases. Then based on the strategy selection parameters, such as observation mode, reusability flag, etc., the suitable Strategy will be selected for the retrieval of a particular Target Scene. However, it is also possible to build the strategies dynamically. Ideally, we would like to have an intelligent strategy selection system that selects the best values for every parameter of the algorithm.

In TES Strategy Builder software, the strategies have the same parameter list in every case. Therefore, here, we are dealing with a static parameter list. However, as it was mentioned earlier, a dynamic parameter list is more desirable. The mentioned alternatives to the TES approach will be discussed later in this paper.

Sequence Level Strategy and Step Level Strategy

As mentioned earlier, the complete retrieval of the atmospheric constituents is done by a set of sequential retrievals. Analysis of the retrieval parameters and the sequence of the retrievals and their relationships with each other indicates that there could be two separate sets of strategy parameters in the overall Retrieval Strategy. 1) The set of the parameters that can be specified for the entire sequence of retrievals and their values remain the same for every individual retrieval. 2) The set of parameters whose values can be different for each individual retrieval.

To make it clear, consider the example of the sequence level parameters in TES case. The retrieval sequence is the parameter that specifies the number and order of sequential retrievals as well as the combinations of the retrieval elements. These values are generic and remain unchanged throughout the sequence, whereas the values of the Step Strategy parameters may differ from one step to another.

Data Flow Diagrams

The diagram in figure 6 expands the Retrieval Strategy and Input Building process in Figure 1. one level down. As shown in this diagram, the first bubble will specify the values for the strategy parameters that are generic for all the steps of the sequence. These values are called the Sequence Level Strategy. In TES case and probably many other cases, these include the retrieval elements arrangement for the entire sequence. This arrangement is shown as the second column in the strategy table shown previously. Once this arrangement is known, at each step, the elements to be retrieved are known. Then the value of other parameters such as the Initial Guess, A Priori data (as shown in the strategy table) can be specified for the identified retrieval elements for a particular Retrieval Step. The collection of these values is called the Step Level Strategy. Then the Get Input process can build the appropriate input for the Retrieval process by using the Step Level Strategy. The results of the Retrieval process will then be analyzed and the loop decides to quit or move on to building strategy for the next Retrieval Step. If the loop decides to continue, then the Get Strategy process will build the Strategy for the next Retrieval Step, knowing about the Retrieval Step that was just done and what comes next by looking at the Sequence Level Strategy that was built in the beginning of the loop.

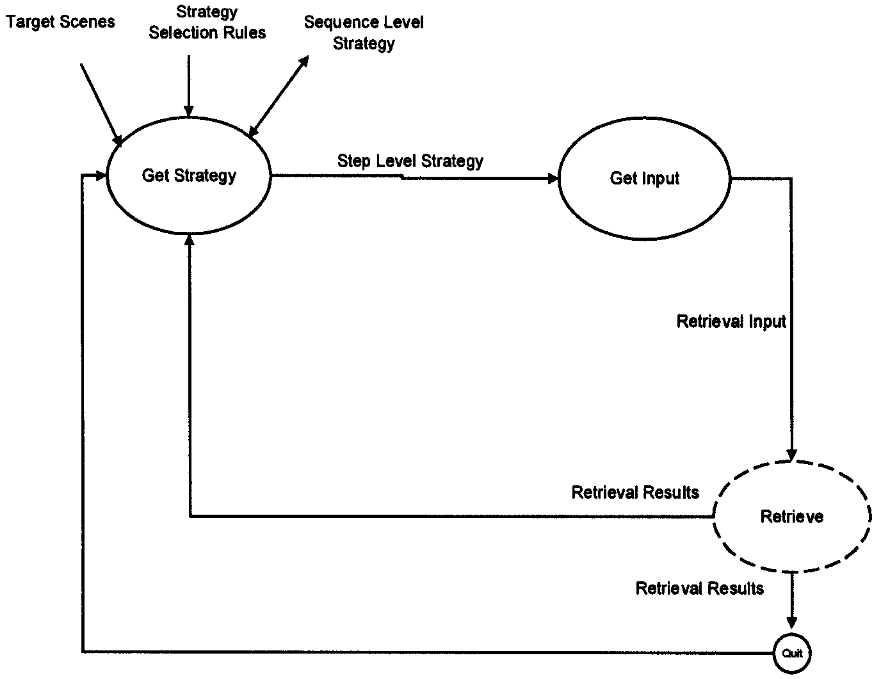


Figure 6 - Strategy Building DFD

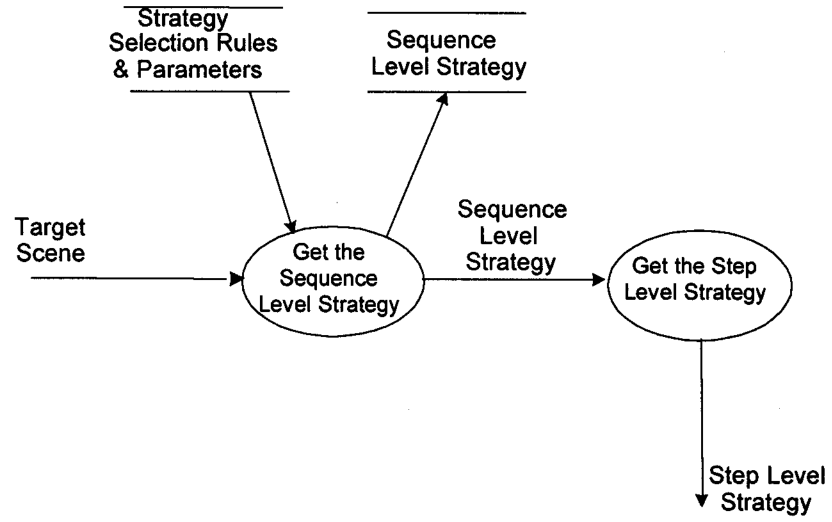

Strategy Selection Rules & Parameters
Sequence Level Strategy
Target Scene
Get the Sequence Level Strategy
Sequence Level Strategy
Get the Step Level Strategy
Step Level Strategy

Figure 7 - Get Strategy Data Flow Diagram

The diagram figure 7 expands the Get Strategy process. During this process, the tasks of specifying the Sequence Level strategy building, and Step Level Strategy building for every Step Strategy are carried out.

As it is shown in the diagram, the Sequence Level Strategy building will be done by using the Target Scene characteristics and Spectrum data as well as Strategy Selection Rules. The Strategy Selection Rules are a set of rules and logic which the "Get Sequence Level Strategy" process uses to choose the appropriate Sequence Level and/or Step Level strategies. An example of a rule for specifying a Step Level Strategy would be the rule which determines which source is the best to use as an Initial Guess source for retrieving H_2O in a particular Retrieval Step (e.g., Col. 4, Row 2 of table in figure 5).

The diagram in figure 8 shows the Strategy Builder with the sequence loop. It shows the major components of the software for building strategy and input for one L1B spectrum data of one target scene.

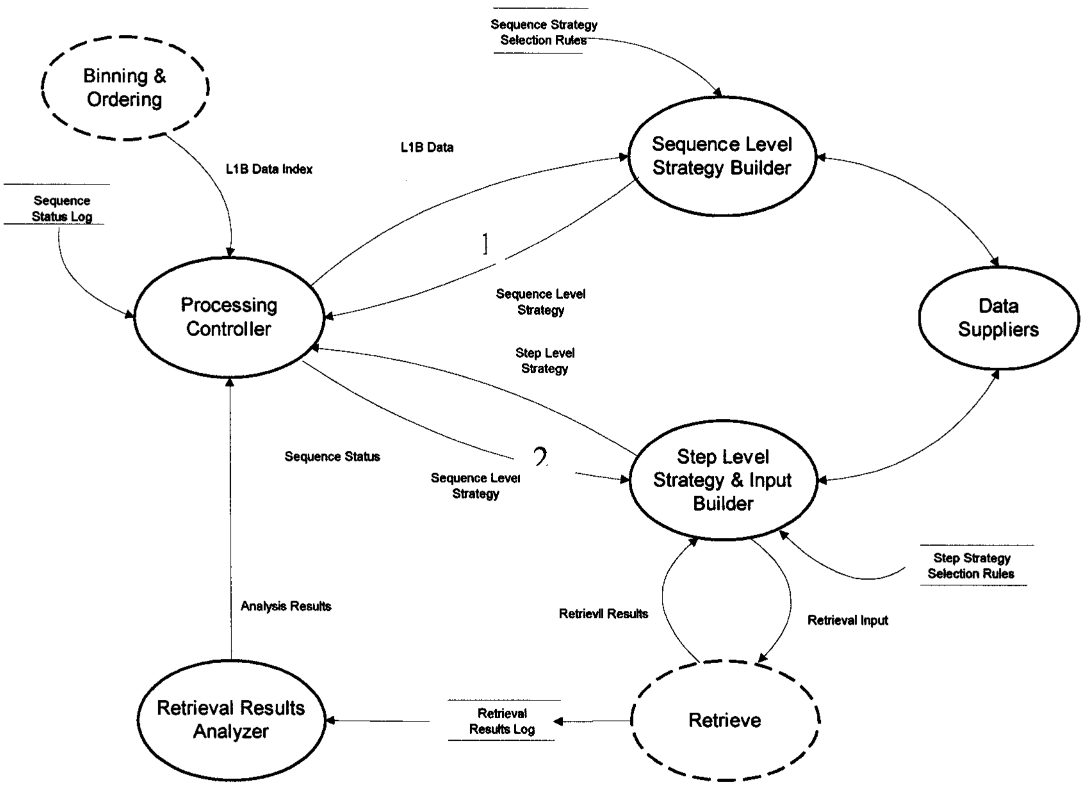


Figure 8 - Strategy Builder with the sequence loop Data flow diagram

When starting the processing of one target scene, the controller uses the Sequence Level Strategy Builder to build the sequence level parameters. Once the sequence level parameters are built, the controller will use the Step Level Strategy Builder to initialize the strategy parameters for the first retrieval in the sequence of retrievals. The Step level Strategy Builder will also build the Retrieval's input and pass it to the Retrieval Engine. Once the Retrieval is done, the results will be passed back to the Step Level Strategy Builder and then to the controller. By analyzing these results, the controller will decide to continue the sequence of the retrievals or to quit the sequence for this target scene and start processing the next target scene. In the first case, it will take the path number 2 to build a strategy and get the input object for the next retrieval in the sequence. In the other case, after getting the next target scene (L1B spectrum), it will start on path 1 again. The flowchart in figure 9 shows the conditions on which the different paths are taken as described.

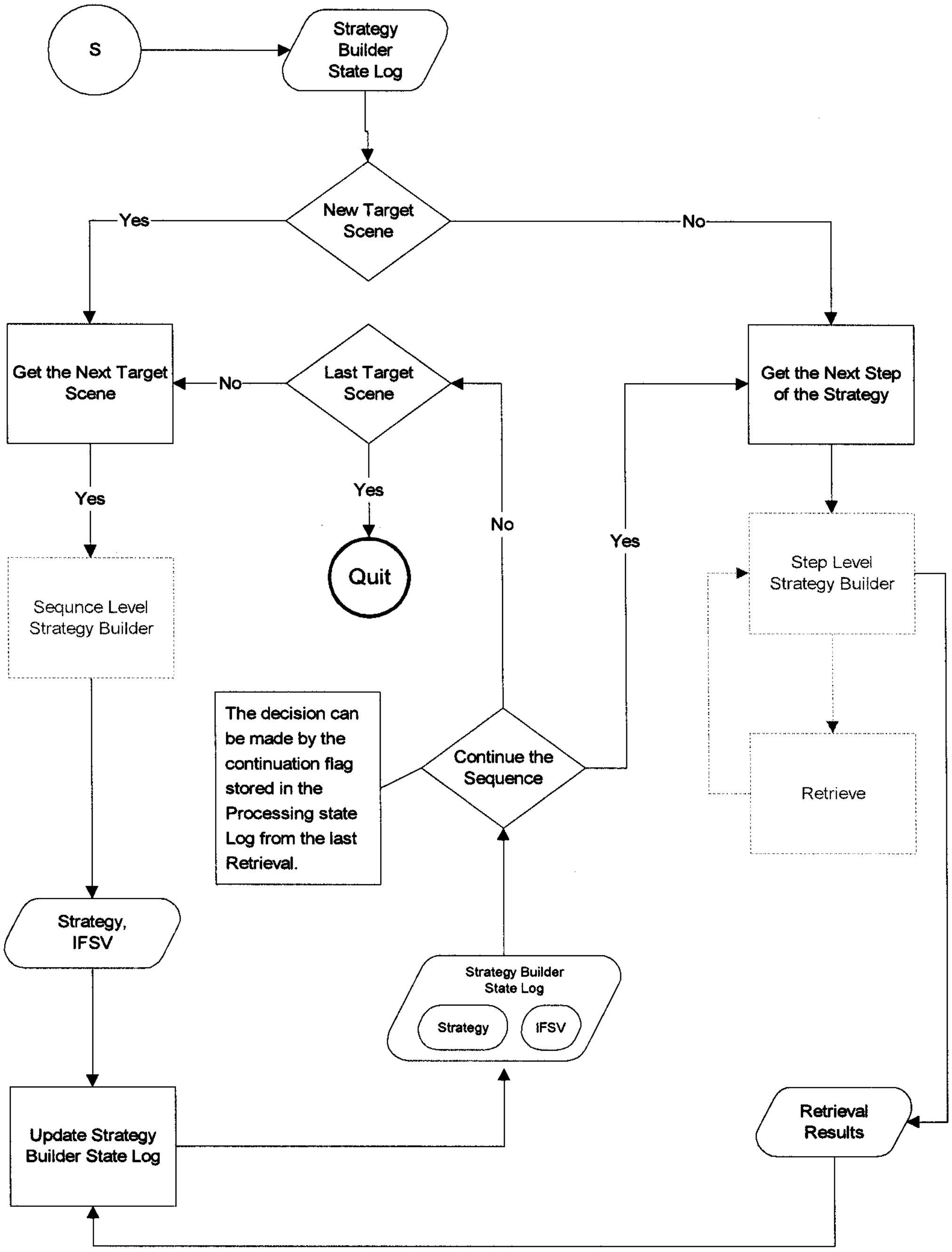

S
Strategy Builder State Log
New Target Scene
Yes
No
Get the Next Target Scene
No
Last Target Scene
Yes
Quit
Get the Next Step of the Strategy
Yes
Sequnce Level Strategy Builder
No
Yes
Step Level Strategy Builder
Retrieve
The decision can be made by the continuation flag stored in the Processing state Log from the last Retrieval.
Continue the Sequence
Strategy, IFSV
Strategy Builder State Log
Strategy
IFSV
Update Strategy Builder State Log
Retrieval Results

Figure 9 - Controller Flowchart

Figures 9 and 10 are the data flow diagrams for the Sequence-Level and Step-Level Strategy Builders.

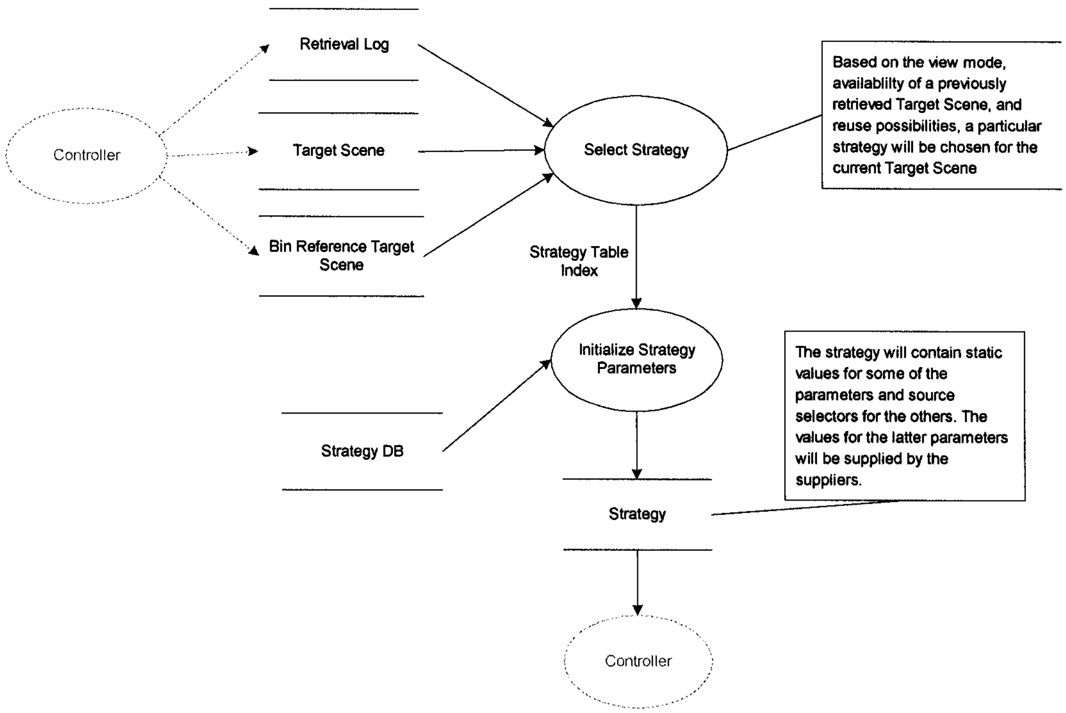
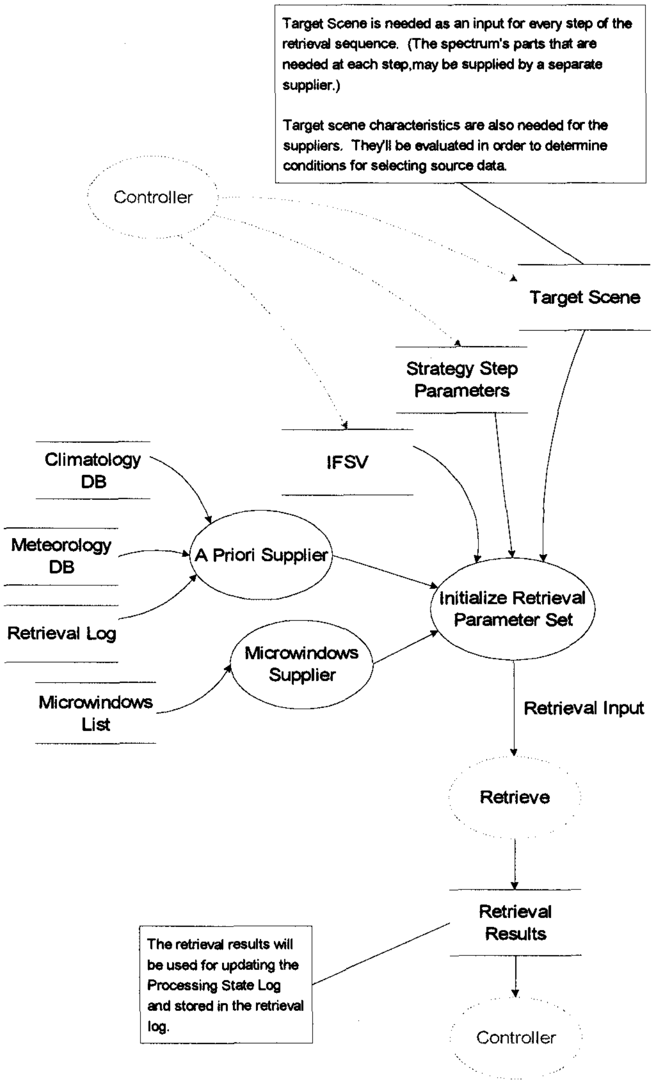

Retrieval Log
Controller
Target Scene
Bin Reference Target Scene
Select Strategy
Based on the view mode, availablilty of a previously retrieved Target Scene, and reuse possibilities, a particular strategy will be chosen for the current Target Scene
Strategy Table Index
Initialize Strategy Parameters
The strategy will contain static values for some of the parameters and source selectors for the others. The values for the latter parameters will be supplied by the suppliers.
Strategy DB
Strategy
Controller

Figure 10 - Sequence Level Strategy Builder


Target Scene is needed as an input for every step of the retrieval sequence. (The spectrum's parts that are needed at each step,may be supplied by a separate supplier.)

Target scene characteristics are also needed for the suppliers. They'll be evaluated in order to determine conditions for selecting source data.
Controller
Target Scene
Strategy Step Parameters
IFSV
Climatology DB
Meteorology DB
Retrieval Log
Microwindows List
A Priori Supplier
Microwindows Supplier
Initialize Retrieval Parameter Set
Retrieval Input
Retrieve
Retrieval Results
The retrieval results will be used for updating the Processing State Log and stored in the retrieval log.
Controller

Figure 11 - Step Level Strategy Builder

5. OBJECT DESIGN

In this section high-level object design of major concepts of Strategy Builder is presented. Object Models with UML notations [2] are made using the Rational Rose 2000 CASE tool.

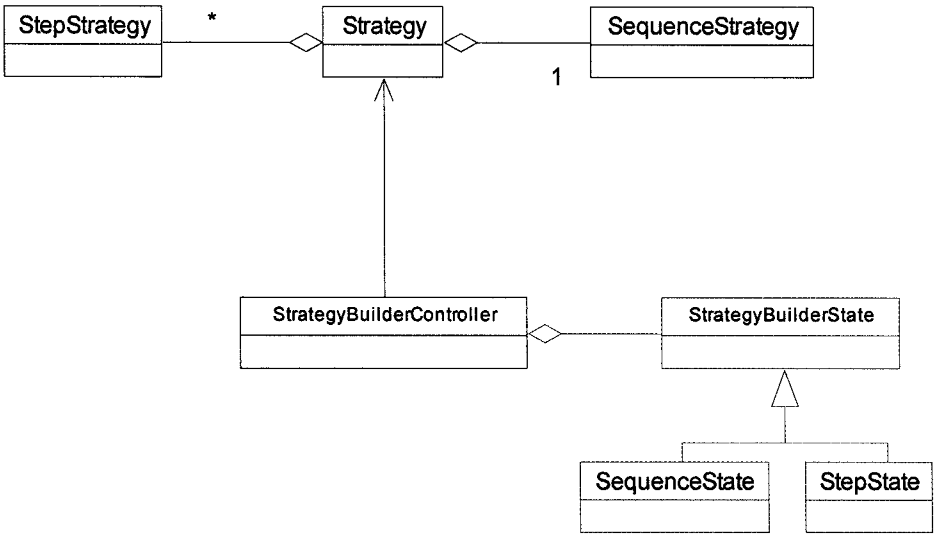


Figure 12 - Strategy and Input Building High-Level Class Diagram

The diagram in figure 12 shows that Strategy consists of Sequence Strategy and Step Strategy. This diagram shows the structure of the Strategy as a whole. It consists of the strategy for all steps of the retrieval. The entire strategy object could be useful for results analysis.

This diagram also shows the StrategyBuilderController class which changes state during the sequence of the retrievals. For this reason, the StrategyBuilderState class will help the controller to have different behaviors in different situations.

When the internal state of the controller object changes, we would like to have the methods behave differently. For example if the Sequence Strategy is not built yet, we would like some buildStrategy method build the Sequence Strategy, but once it is built we would like the same method build the next Step Strategy. In other words different implementations are needed for different situations depending on the state of the controller object. The state class for the controller helps the methods in the controller class behave according to the state of the strategy building. State class uses the State design pattern [1]. The StrategyBuilderState class has two subclasses, SequenceState and StepState, which implement state-specific behavior. The StrategyBuilderState abstract class has the same interface for all the classes that represent different states, i.e. StepState and SequenceState. Here, we can take advantage of polymorphism to choose the method in the appropriate class. Depending on an object of which of these two subclasses (SequenceState or StepState) is instantiated by the controller class, the methods in that class will be invoked.

Figure 13 shows the class structure of the Step Strategy and its components. As shown in this diagram, the Step Strategy, which is a part of the Strategy class consists of several objects of the StepStrategyParameter class. The Step Strategy Parameter class is a generalized class with the individual parameters as its subclasses. InitilGuess, Apriori, and the MicrowindowList classes are three examples of the atmospheric retrieval parameters.

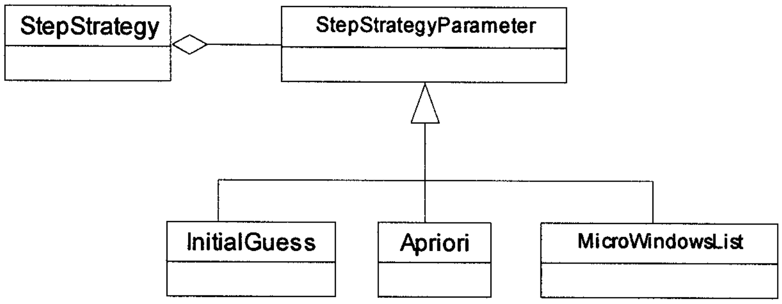


Figure 13 - Step Strategy High-Level Class Diagram

The diagram in figure 14 is the high-level class diagram for an example parameter. Based on the need of the software a supplier class can carry out the task of supplying data for a particular strategy parameter such as Initial Guess based on the required parameter. Examples of the required parameters for Initial Guess selection methods are the species and source. In this diagram the suppliers subclasses of the DAO and ECMWF meteorology sources are shown as examples. A parameterized Factory method [1] may be used to instantiate any of these subclasses as needed.

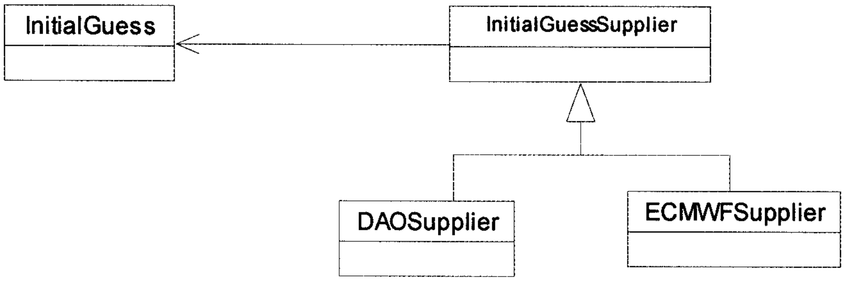


Figure 14 - Example Supplier Class Diagram

The high-level structure of the Retrieval Input class is shown in figure 15. Since the retrieval input consists of other components other than what the strategy provides, the Retrieval Input aggregate three classes as shown. As it is shown here for building the input we will need to initialize the Step Strategy as well as other components.

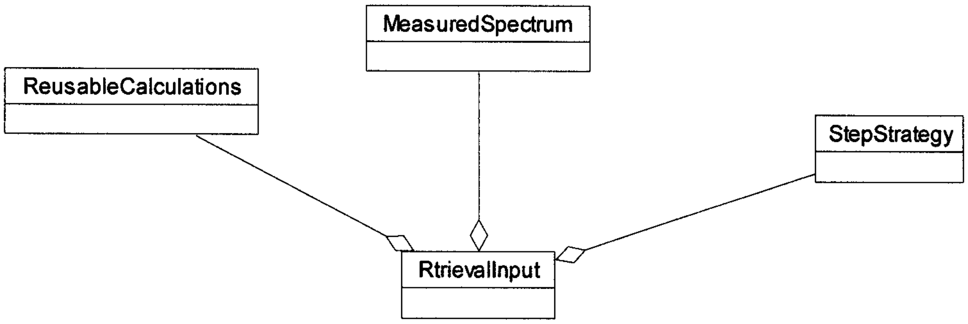


Figure 15 - Retrieval Input class diagram

Strategy Selection is a major part of Strategy building. The diagram in figure 16 shows the strategy selection classes with their relationships. The design used here allows each parameter have its own strategy selection logic. This design also allows dynamic strategy selection. There is a parallel structure between the strategy parameters with their selector classes.

Please note that in the following diagram, the inheritance hierarchy is only shown for the StepStrategySelectionParameters class. The same hierarchy may exist for SequenceStrategySelectionParameters class as well. In this diagram only the Step Strategy related hierarchy is shown for simplicity.

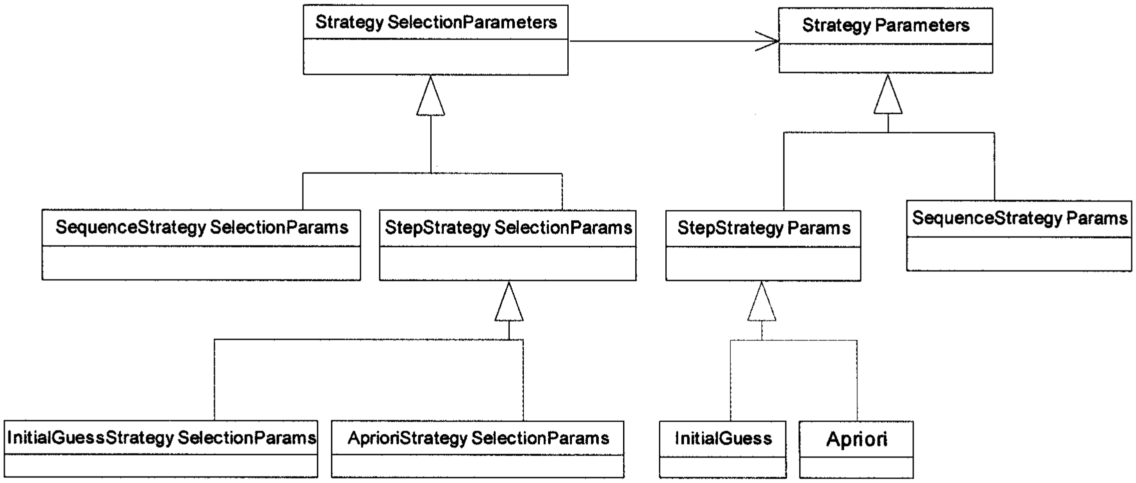

Strategy SelectionParameters
Strategy Parameters
SequenceStrategy SelectionParams
StepStrategy SelectionParams
StepStrategy Params
SequenceStrategy Params
InitialGuessStrategy SelectionParams
AprioriStrategy SelectionParams
InitialGuess
Apriori

Figure 16 - Strategy Selection Class Diagram

6. CONCLUSION

Modifiability is an important feature of retrieval software. It allows increase in the research software's lifetime. Development of Strategy Builder for the atmospheric retrieval software provides the modifiability it needs. Particularly, object-oriented approach in the development of this software makes achieving this goal easier. In addition, it increases its reusability in its domain. Research projects developing retrieval algorithms may use concepts and designed presented to achieve more robust and cost effective retrieval software. More details on the Strategy Builder software, especially on its domain analysis, and architectural, high-level and detail designs, can be found in the author's Graduate Project paper.

Design of the Strategy Builder allows the values for retrieval parameters be selected dynamically if necessary. This flexibility makes the implementation of a learning system such as analyzer possible. The design of the analyzer is not discussed in this paper and is left for future work.

The design presented encompasses several common concepts in atmospheric retrieval software such as potential Retrieval parameters and sequential Retrievals. Specific project can adapt the design to the level of extensibility that the project needs and still benefit from the flexibility that it provides to their atmospheric research software.

REFERENCES

[1] Gamma, Helm, Johnson, Vlissides, Design Patterns, Elements of Reusable Object-Oriented Software, Addison-Wesley, June 1999

[2] Quatrani, T., Visual Modeling with Rational Rose and UML, July 1998

[3] Bass, Clements, and Kazman. Software Architecture in Practice, Addison-Wesley 1997

[4] Mary Shaw, David Garlan, Carnegie Mellon University. Software Architecture, Perspectives on an emerging discipline, Prentice-Hall, Inc. 1996

[5] Roger S. Pressman, Software Engineering, A practioner's approach, McGraw-Hill 1997

[6] TES Algorithm Theoretical Basis Document, Version 1.1, Jet Propulsion Laboratory Internal Document D-1674, October 1, 1999

[7] MIPAS High level algorithm definition and physical and mathematical optimizations, Issue 2, Jet Propulsion Laboratory, August 1997

[8] HIRDLS Algorithm Theoretical Basis Document, Version 2, Jet Propulsion Laboratory October 1999

[9] Firesmith. D. G., Object-Oriented requirements Analysis and logical Design, John Wiley & Sons 1996